ELETTRONICA DIGITALE

Prof. Ing. Paolo Visconti

Facoltà di Ingegneria - Università del Salento

Obiettivi del Corso

Il corso costituisce la base per lo studio ed il progetto dei sistemi elettronici digitali. Vengono fornite allo studente le metodologie di analisi e di progetto dei circuiti digitali combinatori e sequenziali ed illustrati i principi di funzionamento, prestazioni e limiti delle famiglie logiche e dei principali circuiti elettronici utilizzati nell'elaborazione numerica di dati e segnali.

Requisiti

Si richiede una buona conoscenza dei principi di funzionamento e delle caratteristiche dei principali dispositivi allo stato solido (diodi a giunzione, transistor BJT, JFET e MOSFET) nonché dei più comuni metodi di soluzione delle reti elettriche.

Programma dettagliato del Corso

- Introduzione ai sistemi digitali
  - Sistemi digitali, segnali logici ed analogici, vantaggi dei sistemi digitali, rappresentazione binaria, algebra di Boole, porte logiche elementari, funzioni logiche, mappe di Karnaugh e sintesi di funzioni logiche, alee statiche in reti combinatorie.
- Reti combinatorie con uscite multiple.
  - Decodificatori, codificatori, multiplexer e demultiplexer digitali, comparatori digitali, sommatore ad n bits, rilevatori e generatori di parità.
- Famiglie logiche
  - Introduzione, definizione dei parametri (corrente, tensione) di ingresso/uscita di una porta logica, margine di rumore, potenza dissipata, caratteristica di trasferimento. Famiglia logica DL (Diode Logic), DTL (Diode Transistor Logic), TTL (Transistor-Transistor Logic), ECL (Emitter Coupling Logic), famiglie logiche unipolari (NMOS, Pseudo-NMOS, Complementary MOS – CMOS), BiCMOS (Bipolar CMOS).
  - Porte logiche TTL e CMOS Three-State, Wired-Logic, Interfacciamento tra porte logiche di famiglie diverse.

- Circuiti sequenziali
  - Introduzione, circuiti logici con memoria, Latch di porte NAND, Latch di porte NOR, segnale di clock e Flip-Flop cadenzati, FF SC cadenzato, FF JK cadenzato, FF D cadenzato, FF Master-Slave.
  - Registri: introduzione, a scorrimento, registri MOS, trasferimento dati parallelo e seriale tra registri. Contatori: caratteristiche generali. Contatori asincroni (modulo 8, modulo 16, decimale), contatore a decremento, contatori binari sincroni, ad anello, contatore di Johnson.

Esercitazioni durante il Corso

- Circuiti combinatori: risoluzione di esercizi d'esame di tipo combinatorio.
- Famglie logiche: risoluzione di esercizi d'esame sulle famiglie logiche. Progetto e dimensionamento di porte TTL e CMOS.
- Potenza dinamica dissipata e ritardi di propagazione. Analisi di circuiti combinatori-sequenziali per il calcolo della potenza dinamica dissipata e del ritardo di propagazione.
- Circuiti sequenziali: risoluzione di esercizi d'esame sui circuiti sequenziali (Flip-Flop, registri, contatori).

Sistemi Digitali

- Un sistema digitale è un qualsiasi sistema elettronico in cui le informazioni vengono rappresentate in **forma binaria** e l'elaborazione si basa sull'**algebra di Boole**
- Al giorno d'oggi praticamente qualsiasi sistema elettronico è per la maggior parte digitale a partire (ovviamente) dal computer fino ad arrivare ai telefonini, i lettori CD, DVD e (in futuro) gli elettrodomestici passando per l'ABS, il sistema di controllo dell'airbag e via dicendo.

I principali vantaggi dei sistemi digitali sono:

- Riproducibilità dei risultati
- Semplicità di progettazione
- Flessibilità
- Programmabilità
- Velocità di elaborazione
- Costo

BLOCCHI FUNZIONALI

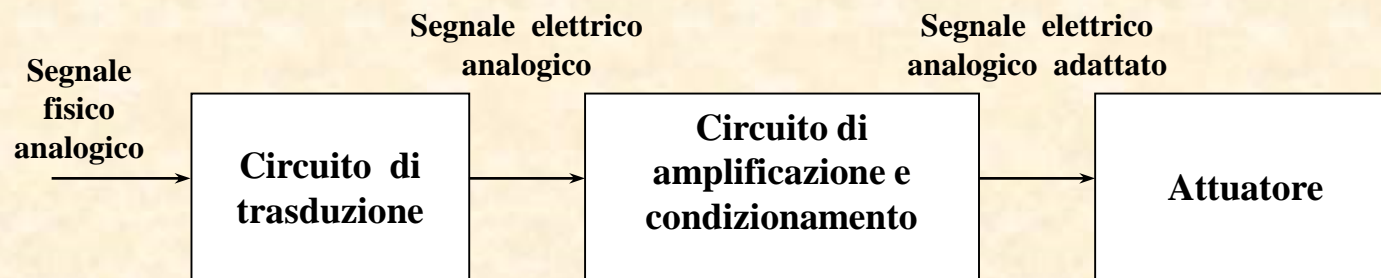
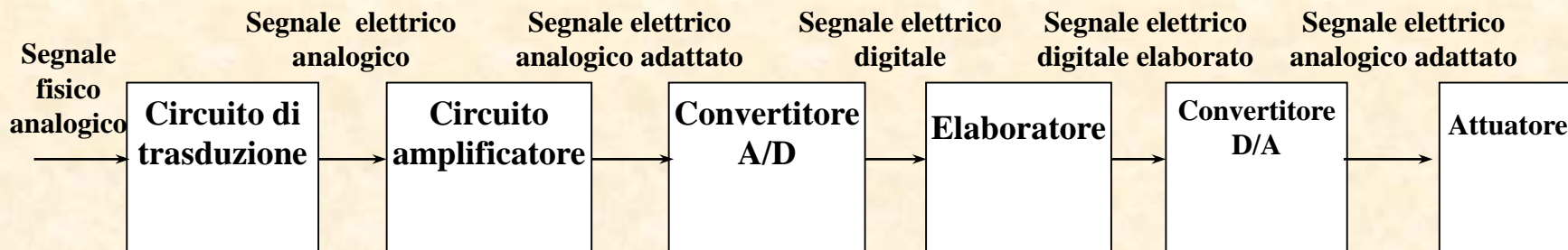
- I sistemi elettronici sono presenti in maniera pervasiva in tutti i settori applicativi di una società avanzata: sistemi informatici (computer), sistemi di telecomunicazione (radio, televisione, telefono, satelliti, GPS, ...), sistemi di produzione (automazione, controllo dei processi), mezzi di trasporto (controllo della trazione, sistemi di sicurezza, strumentazione, ...), ecc.
- I sistemi elettronici sono generalmente scomponibili in sottosistemi o blocchi funzionali, a loro volta costituiti da uno o più circuiti elettronici. Questi ultimi contengono componenti passivi (resistori, induttori, condensatori, diodi) ed attivi (transistori e amplificatori operazionali).

BLOCCHI FUNZIONALI

- AMPLIFICATORI: aumentano la potenza di segnali deboli.
- FILTRI: separano segnali desiderati da segnali indesiderati e rumore.
- GENERATORI DI SEGNALE: generano segnali di diverse forma d'onda (es: sinusoidi, onde quadre, ..).
- CIRCUITI DI WAVESHAPING: cambiano la forma d'onda di un segnale in un'altra.
- CIRCUITI LOGICI: eseguono funzioni logiche su segnali digitali.
- MEMORIE DIGITALI: immagazzinano informazioni sotto forma di segnali digitali.
- ALIMENTATORI: forniscono la potenza DC necessaria a tutti i blocchi funzionali.
- CONVERTITORI A/D e D/A: convertono segnali analogici in segnali digitali e viceversa.

Esempio di sistema elettronico: sistema di controllo

Un sistema di elaborazione può essere rappresentato mediante schemi a blocchi, connessi in cascata, le cui funzioni dipendono dal tipo di segnale da elaborare.

Un sistema ***analogico*** contiene dispositivi e circuiti che manipolano segnali di tipo fisico o elettrico in forma analogica, cioè quantità variabili in un intervallo continuo di valori.

Un sistema ***digitale*** è una combinazione di dispositivi e circuiti per manipolare segnali in forma analogica e digitale, cioè quantità che possono assumere solo certi livelli discreti.

Le apparecchiature elettroniche sono in grado di manipolare segnali di natura elettrica (corrente, tensione); per determinati segnali fisici (temperatura, pressione …) è quindi necessaria una trasformazione in segnali di natura elettrica tramite l'uso di opportuni trasduttori o sensori.

INFORMATION E POWER PROCESSING

- Molti sistemi elettronici (es: calcolatori elettronici, memorie, sistemi di comunicazione, strumentazione, sistemi di controllo) hanno come funzione quella di estrarre, elaborare, trasportare o conservare segnali cui è associata informazione.
- Altri sistemi (es: sistema audio, azionamento elettrico, pacemaker cardiaco, alimentatore, ...) devono trasferire potenza ad un dispositivo utilizzatore: questi sistemi sono interessati al contenuto di potenza delle grandezze elettriche che processano piuttosto che a quello informativo.

Vantaggi delle tecniche e dei sistemi digitali

- I sistemi digitali sono più semplici da progettare. Ciò è dovuto al fatto che i circuiti impiegati sono circuiti di commutazione in cui non sono determinanti i valori esatti di tensione o corrente ma soltanto i valori logici (alto o basso).

- E' facile memorizzare l'informazione mediante dispositivi di memoria che possono registrare l'informazione e mantenerla per tutto il tempo necessario.

- L'accuratezza e la precisione sono maggiori. Infatti i circuiti digitali sono meno affetti dal rumore. Le fluttuazioni spurie della tensione (rumore) non sono critiche nei sistemi digitali poiché il valore esatto della tensione non è importante.

- Il funzionamento dei sistemi digitali può essere facilmente programmato.

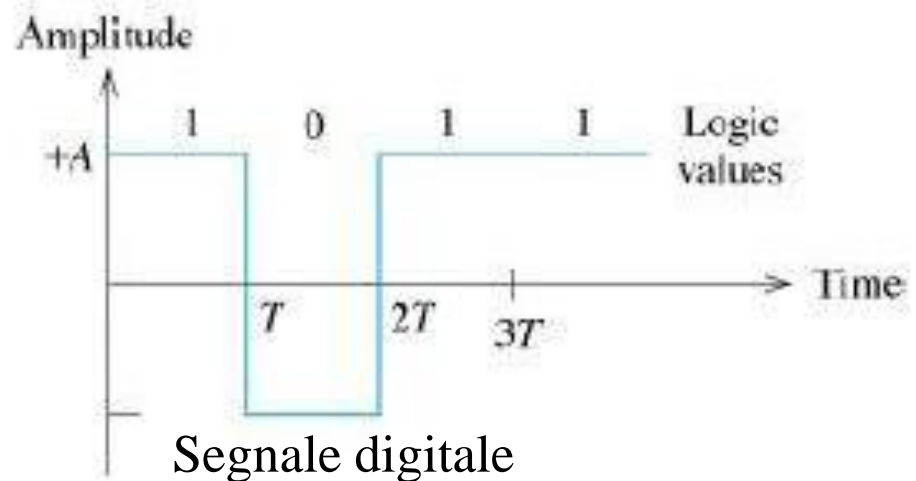
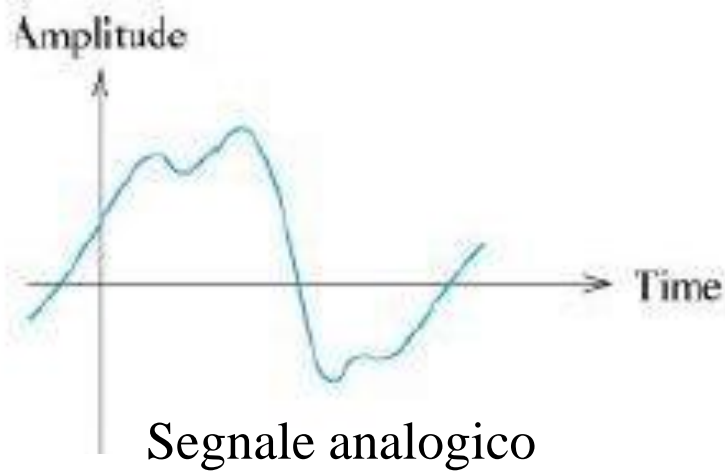
- Più circuiti digitali possono essere fabbricati sullo stesso chip occupando un'area minima con un elevatissimo grado di integrazione.

SVANTAGGIO: ***il mondo reale è essenzialmente ANALOGICO.***

SEGNALI ANALOGICI E DIGITALI

Segnale analogico: il suo valore varia in un intervallo continuo.

Segnale digitale: può assumere soltanto un numero discreto di valori (es: segnali binari assumono due valori, '0' e '1').

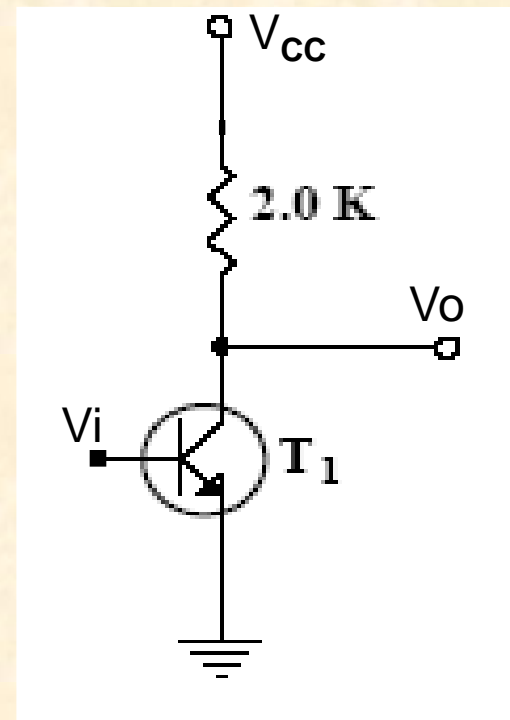
Segnale analogico

Segnale digitale

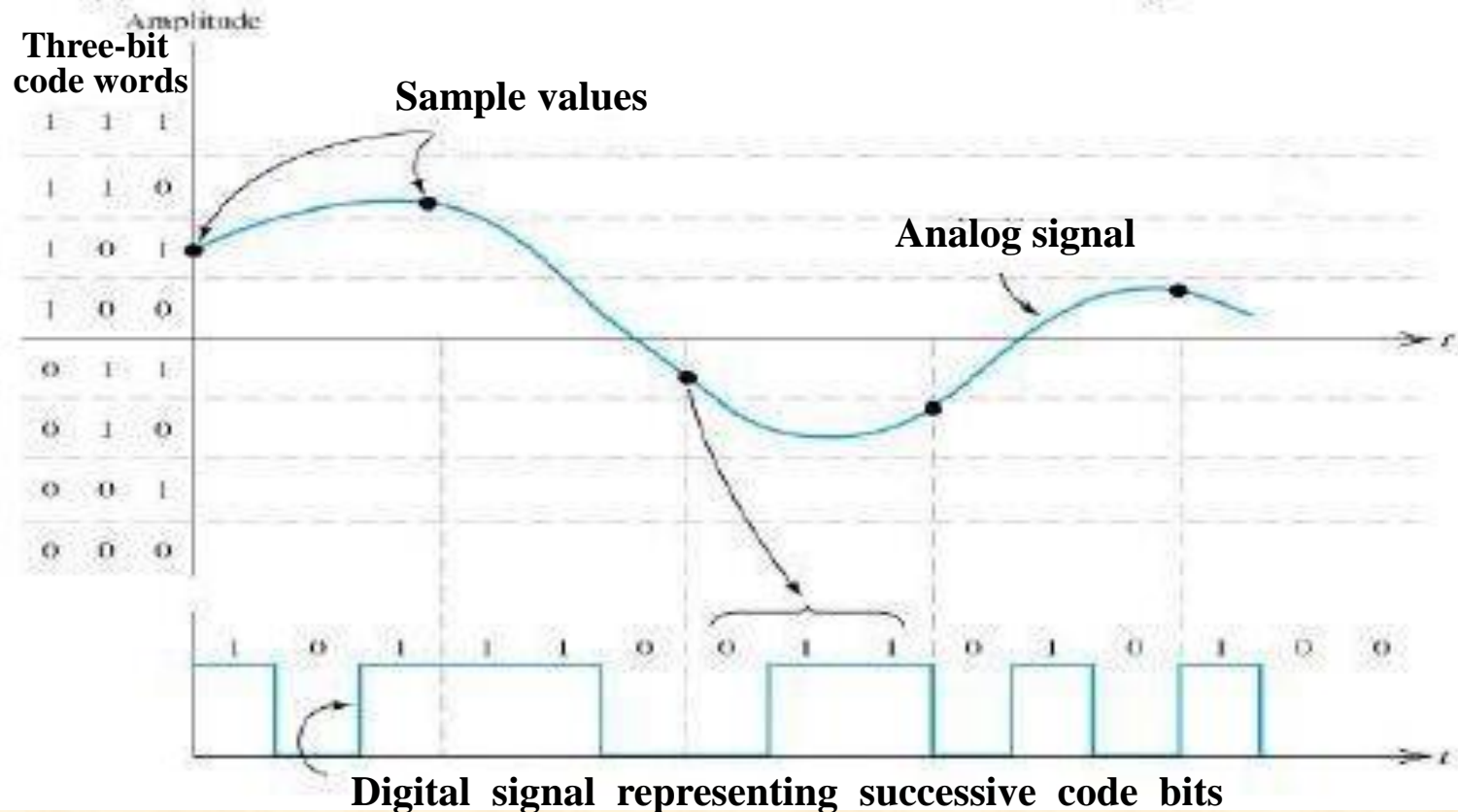
Un segnale si dice ***analogico*** quando può assumere nel tempo, con continuità, tutti i valori compresi fra un minimo e un massimo.

Un segnale si dice ***digitale*** (o logico) quando può assumere nel tempo soltanto alcuni valori compresi fra un minimo e un massimo.

SEGNALI DIGITALI E LIVELLI LOGICI

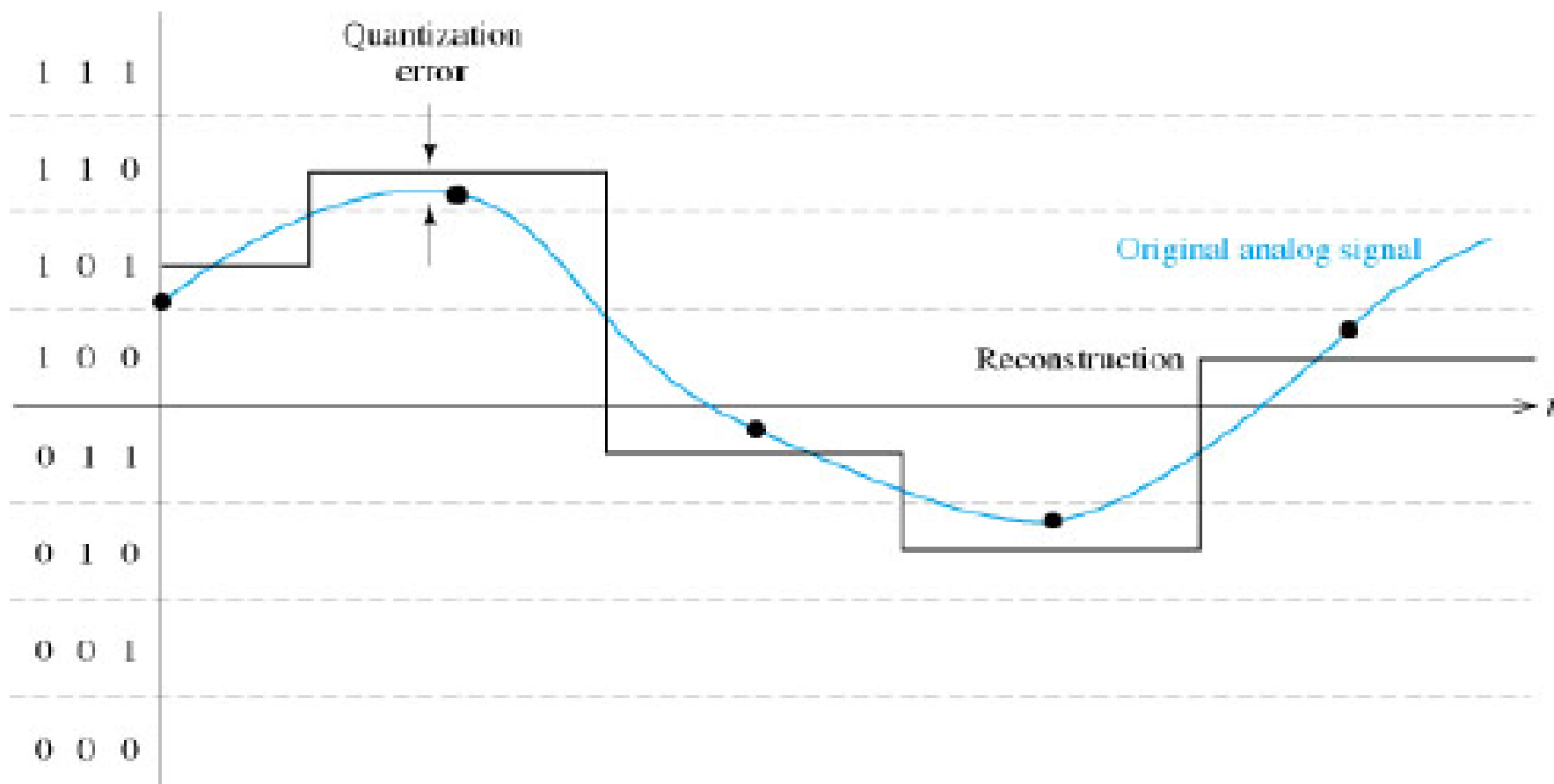
E’ evidente che più elevato è il numero di valori che il segnale logico può assumere allora più vicini risulteranno i livelli di tensione che li rappresentano. Per evitare dunque commutazioni errate, è preferibile usare il numero minimo di livelli in modo che la distanza tra gli stessi risulti la massima; il caso migliore è perciò quello di 2 soli valori.

Vi è un’altra considerazione che avvalora questa scelta: i circuiti logici sono realizzati utilizzando componenti attivi elettronici (BJT, MOSFET ecc). Dunque è possibile far lavorare i transistors nelle zone di intedizione o saturazione e non in zona attiva, come invece succede prevalentemente nel caso di circuiti analogici. Possiamo indicare con 0 logico lo stato del transistor bipolare (BJT T_1) nella zona di saturazione (Vo = $V_{CESAT} \approx 0{,}2$ V), mentre con 1 indichiamo lo stato nella zona di interdizione (Vo = $V_{CE} \approx V_{CC}$) .

CONVERSIONE ANALOGICO-DIGITALE

- La conversione di un segnale analogico in forma digitale è un processo che richiede due passi, detti di campionamento e quantizzazione.
  - *Campionamento*: il segnale analogico viene misurato ad istanti di tempo periodici.
  - *Quantizzazione*: al segnale campionato viene assegnato una parola di codice che rappresenta l'intervallo di valori al cui interno cade il campione.

CONVERSIONE ANALOGICO-DIGITALE

- La frequenza di campionamento dipende dal contenuto frequenziale del segnale da convertire: se il segnale non contiene componenti di frequenza superiore a f_H, il segnale può essere perfettamente ricostruito a partire dai suoi campioni se la frequenza di campionamento è scelta maggiore di $2 \times f_H$!
- Una volta volta effettuata la quantizzazione, l'esatta ampiezza del segnale non può più essere ricostruita, perché tutti i valori che cadono nello stesso intervallo vengono rappresentati dalla stessa parola di codice.
- Il livello associato alla parola di codice è a metà di ogni regione. L'errore che si commette quando si ricostruisce il segnale analogico originario è detto errore di quantizzazione.

CONVERSIONE ANALOGICO-DIGITALE

- L'errore di quantizzazione è tanto minore quanto maggiore è il numero degli intervalli, N. Tanto più grande è N, tuttavia, tanto maggiore è il numero di bit, K, necessario per rappresentarli. Infatti, $N=2^K$
- Esempio: nei CD audio si utilizzano parole a 16 bit.

SISTEMI ANALOGICI vs SISTEMI DIGITALI

Uno dei vantaggi principali dei sistemi digitali su quelli analogici è che il rumore può essere tolto completamente da un segnale digitale, a patto che l'ampiezza del rumore sia inferiore a 1/2 della distanza tra due livelli consecutivi. Lo stesso non può farsi a partire da un segnale analogico in quanto tutti i suoi valori sono validi. Inoltre il rumore tende ad aumentare ad ogni operazione compiuta sul segnale nei sistemi analogici.

SISTEMI ANALOGICI vs SISTEMI DIGITALI

- I circuiti analogici richiedono generalmente meno componenti di quelli digitali: ciò si traduce in un minor costo dei circuiti discreti analogici (cioè realizzati assemblando componenti prodotti separatamente).
- I circuiti digitali si prestano però ad essere realizzati in forma integrata più facilmente di quelli analogici (resistenze, capacità ed induttanze sono infatti difficilmente integrabili soprattutto se di valore elevato). Un circuito integrato è un circuito costituito da componenti e relative interconnessioni prodotti contemporaneamente su uno stesso pezzo (chip) di semiconduttore. Il costo di un circuito integrato non è proporzionale al numero di componenti in esso contenuti. Se prodotti in grande quantità, i circuiti integrati possono essere a bassissimo costo.
- I sistemi digitali sono generalmente più flessibili di quelli analogici, ovvero più facilmente riconfigurabili per applicazioni anche molto diverse fra loro.
- I segnali di input/output sono quasi sempre analogici. Operazioni su segnali di piccola ampiezza o frequenza molto elevata richiedono un approccio analogico. Sempre più nei moderni sistemi elettronici coesistono blocchi analogici e digitali anche integrati sullo stesso chip.

PROGETTO DEI SISTEMI ELETTRONICI

- Diverse figure professionali concorrono al progetto, alla realizzazione e all'ottimizzazione dei sistemi elettronici:
  - il progettista di sistema definisce il diagramma a blocchi del sistema elettronico;
  - il progettista di circuito definisce la struttura interna dei singoli blocchi circuitali (a livello di componenti e interconnessioni);
  - l'esperto di fisica dei dispositivi studia il funzionamento dei dispositivi elettronici;
  - l'ingegnere di processo si occupa della progettazione e ottimizzazione del processo di fabbricazione dei dispositivi e dei circuiti integrati.

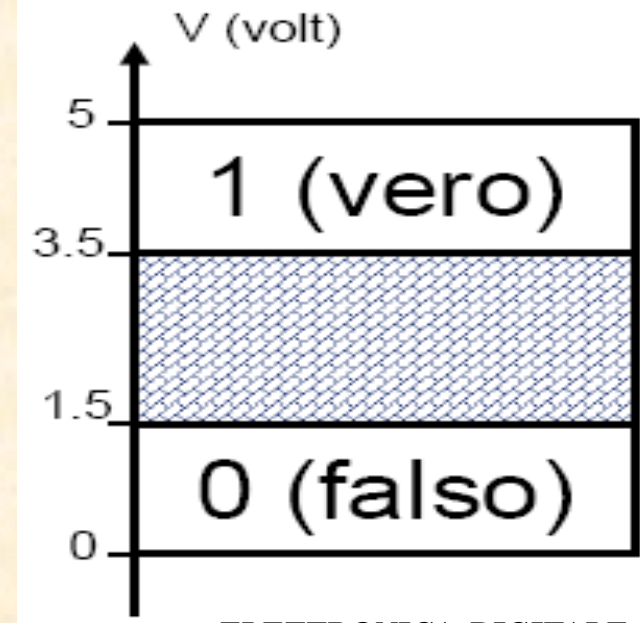
Rappresentazione binaria

- Il segnale una volta quantizzato viene rappresentato in forma binaria, ossia con una notazione posizionale a base 2.
- Questo significa che esistono solo 2 simboli (0,1) con in quali costruire la rappresentazione del numero.

Ma come si rappresentano, fisicamente, i simboli 0 e 1 ?

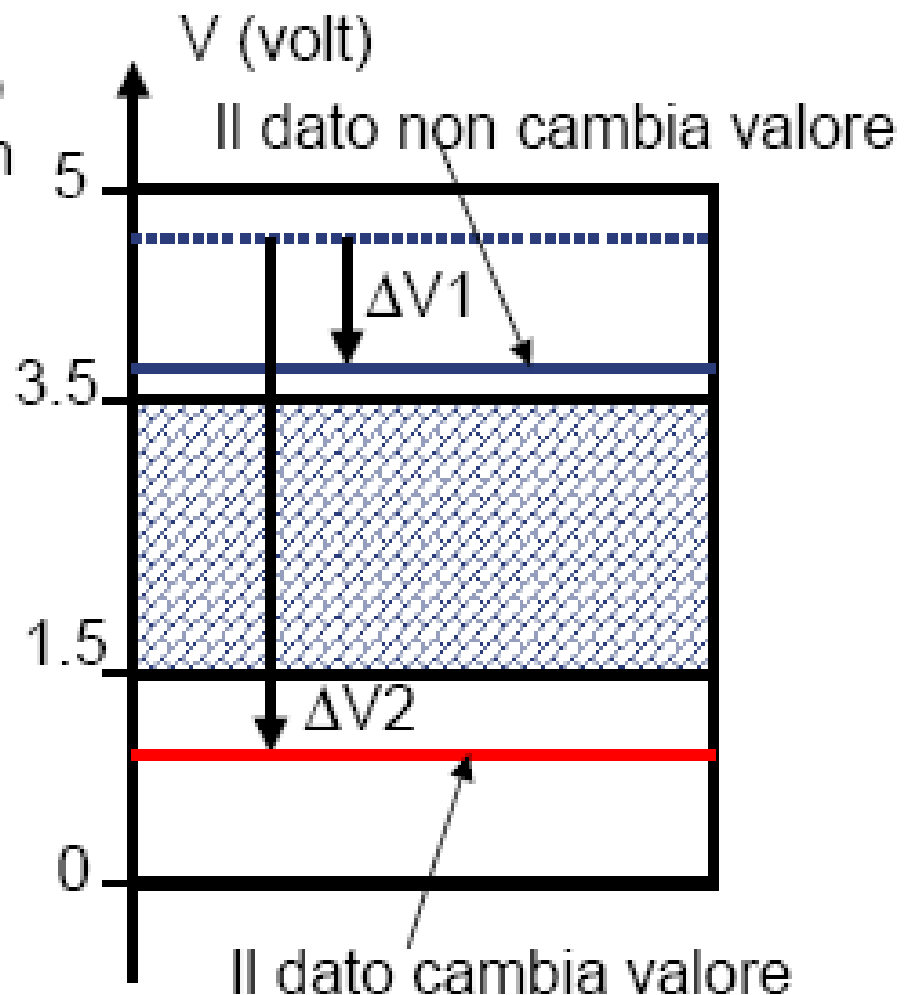
La rappresentazione fisica avviene per mezzo di grandezze elettriche.

A disposizione ci sono fondamentalmente 3 grandezze da utilizzare (tensione, corrente, carica).

Normalmente la scelta cade sulla tensione, che è più facile da maneggiare e misurare.

Rappresentazione binaria

A ciascun simbolo, quindi, viene associato un intervallo di valori di tensione e non un singolo valore.

I due intervalli sono separati da una banda proibita di valori di tensione che non dovrebbero MAI essere raggiunti.

La presenza di disturbi (rumore) quindi non altera i dati a meno che l'intensità del rumore stesso non causi un salto da un intervallo all'altro.

CONCETTI BASE

- La singola cifra binaria (1 o 0) è chiamato **bit** e rappresenta una piccola parte di informazione
- Esempio: R può rappresentare se piove in un posto specifico, per cui R=1 significa che piove e R=0 che non piove
- Per rappresentare più informazioni, si usano gruppi di variabili logiche chiamati **parole digitali: 1 byte** = 8 bit
- Esempio: RWS può rappresentare un insieme di informazioni meteo: R, la pioggia; W, se il vento soffia più di 20 Km/h (in questo caso W=1); S, se c'è il sole (S=1) o è nuvoloso (S=0). La parola '110' indica condizioni di pioggia, vento, con tempo nuvoloso.
- Per analizzare matematicamente le operazioni logiche si usa **l'algebra booleana**

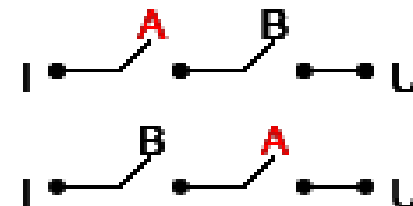
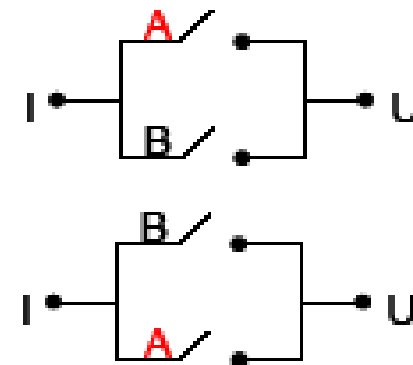
Rappresentazione binaria

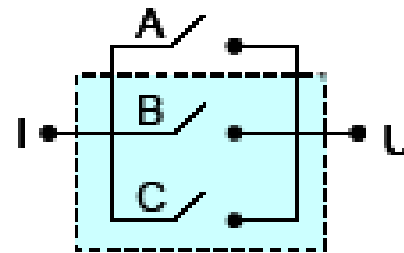
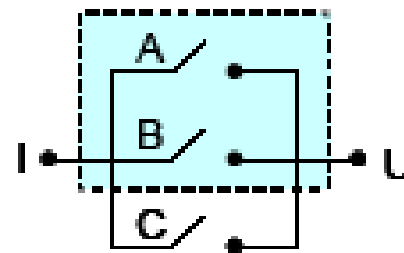
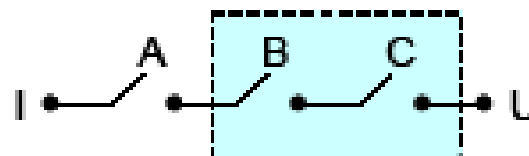
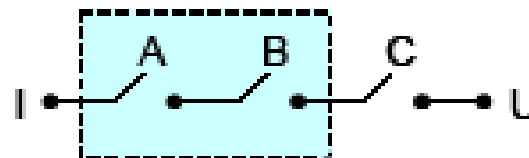
- La rappresentazione binaria è quindi un'astrazione logica che consente di dimenticare i dettagli fisici (il segnale in realtà è una tensione quindi un segnale analogico) per concentrarsi sugli aspetti logici del sistema
- L'esistenza di soli 2 simboli, inoltre, rende più semplice realizzare i circuiti perché il rumore influenza molto meno il comportamento dei dispositivi
- Le operazione logiche seguono le regole dell'algebra di Boole (della commutazione)

- L'algebra di Boole o della commutazione è lo strumento che si usa per l'elaborazione dell'informazione binaria.
- L'algebra di Boole si basa su 2 simboli (0/1) e i 3 operatori: somma (+), prodotto (•) e negazione (').
- Dal corso di calcolatori dovrebbero essere noti gli assiomi ed i teoremi su cui si basa l'algebra della commutazione. In questo corso si farà uso delle nozioni di base come strumento per arrivare alla sintesi di circuiti logici.

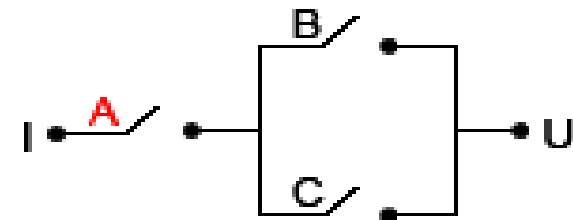
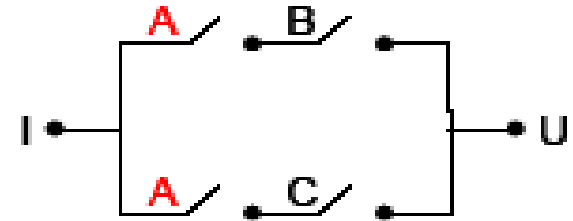
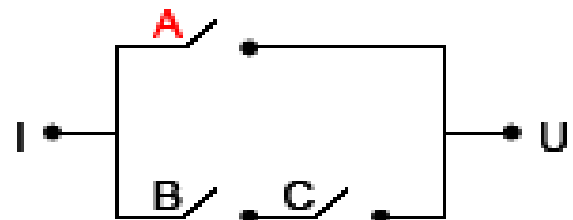
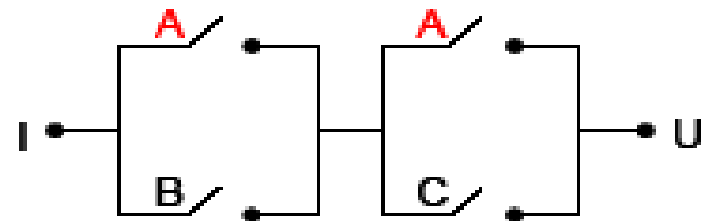
PROPRIETÀ DELL'ALGEBRA BOOLEANA

- Utilizziamo schemi ad interruttori equivalenti: il risultato della funzione logica vale 1 se esiste un collegamento tra ingresso I e uscita U, 0 altrimenti.
- **PROPRIETA' COMMUTATIVA** – Il prodotto e la somma logica di due variabili booleane non cambia se s'inverte l'ordine dei termini.

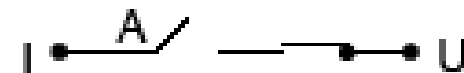
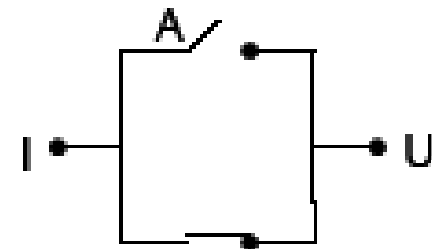
- $A * B = B * A$

- $A + B = B + A$

PROPRIETÀ DELL'ALGEBRA BOOLEANA / 2

- **PROPRIETA' ASSOCIATIVA** – Il risultato dell'operazione non cambia qualunque sia l'ordine con cui l'operazione viene applicata ai termini consecutivi.

  - (A + B) + C = A + (B + C)
  - (A * B) * C = A * (B * C)

PROPRIETÀ DELL'ALGEBRA BOOLEANA /3

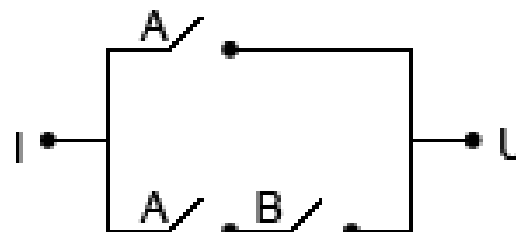
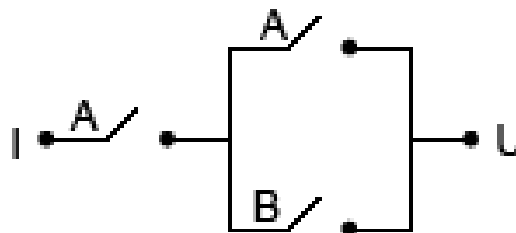
- **PROPRIETA' DISTRIBUTIVA** – Le operazioni di somma e prodotto logico godono della proprietà distributiva.

  - $A * B + A * C = A * (B + C)$

  - $(A + B) * (A + C) = A + (B * C)$

PROPRIETÀ DELL'ALGEBRA BOOLEANA /4

- **COMPLEMENTARIETA'** – Stabilisce che la somma logica di una variabile booleana con il suo complemento è uguale a 1, mentre il prodotto logico tra i due termini è uguale a 0.

  - $A * \bar{A} = 0$

  - $A + \bar{A} = 1$

PROPRIETÀ DELL'ALGEBRA BOOLEANA /5

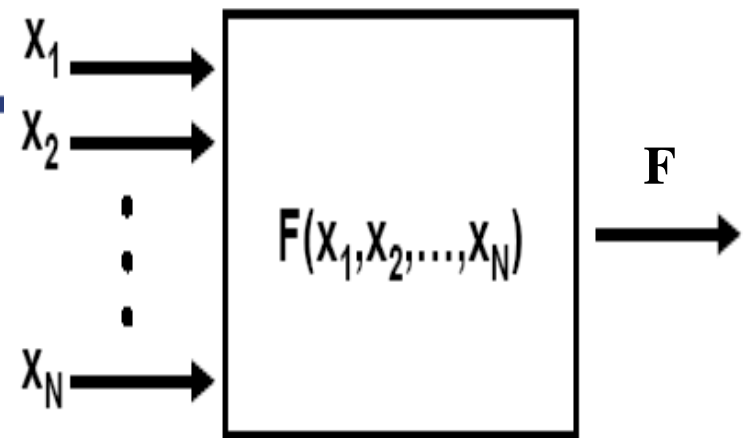
- **ASSORBIMENTO** – La somma di una variabile booleana A con il prodotto tra la stessa variabile ed un'altra (B) è uguale alla variabile A. Il prodotto di una variabile booleana A con la somma tra la stessa variabile ed un'altra (B) è uguale alla variabile A.

- $A + (A * B) = A$

- $A * (A + B) = A$

ALTRE IDENTITÀ BOOLEANE

$$A + 0 = A$$
$$A + 1 = 1$$
$$A * 0 = 0$$
$$A * 1 = A$$
$$\overline{\overline{A}} = A$$

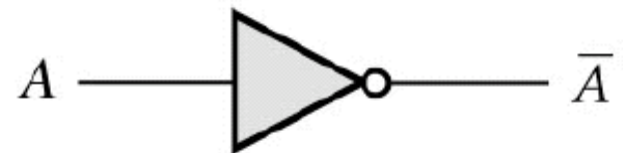
Fare esercizi semplificazione algebrica di funzione logiche 3.13, 3.14, 3.15, 3.16 pag. 83

Funzioni logiche

- Una funzione logica è una relazione algebrica ingresso/uscita che lega un numero N di ingressi con l'uscita.

FUNZIONI LOGICHE: NOT

- **NOT =** funzione ad un ingresso A che fornisce in uscita il suo complemento $\overline{A}$: se A=1, $\overline{A} = 0$, mentre se A=0, $\overline{A} = 1$
- I circuiti che effettuano l'operazione NOT sono chiamati **inverter** (vedi figura: il cerchietto dopo il triangolo sta a indicare che il livello logico dell'uscita è basso)
- Per capire qual'è la funzione logica di un qualsiasi operatore è utile usare la **tabella di verità**, in cui sono riportate le uscite per ciascuna delle **possibili combinazioni dei vari ingressi** (n ingressi => 2^n combinazioni possibili)

A	$\overline{A}$
0	1
1	0

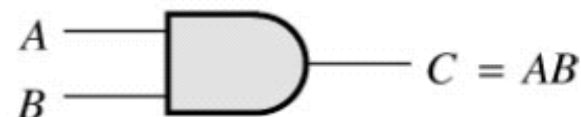
(a) Truth table

(b) Symbol for an inverter

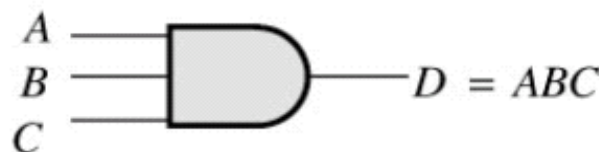
FUNZIONI LOGICHE: AND

- **AND** è un'importante funzione logica, detta anche **prodotto logico** (C=A·B·..), a 2 o più ingressi, che ha **uscita a '1' solo se tutti gli ingressi sono a '1'**
- La tabella verità e il simbolo circuitale della **porta logica AND** (che produce un'uscita uguale a quella della funzione AND) sono riportate in figura

A	B	$C = AB$
0	0	0
0	1	0
1	0	0
1	1	1

(a) Truth table

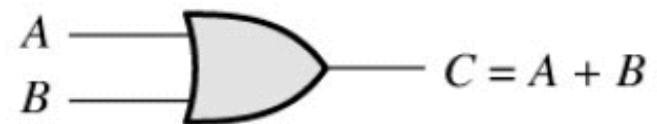
A, B → $C = AB$

(b) Symbol for two-input AND gate

A, B, C → $D = ABC$

(c) Symbol for three-input AND gate

FUNZIONI LOGICHE: OR

- **OR** è un'importante funzione logica, detta anche **somma logica** (C=A+B+..), a 2 o più ingressi, che ha **uscita a '1' se uno degli ingressi è a '1'**
- La tabella verità e il simbolo circuitale della **porta logica OR** (che produce un'uscita uguale a quella della funzione OR) sono riportate in figura
- L'algebra booleana fornisce le regole necessarie a semplificare e minimizzare funzioni logiche complesse

A	B	$C = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(a) Truth table

(b) Symbol for two-input OR gate

NAND, NOR e XOR

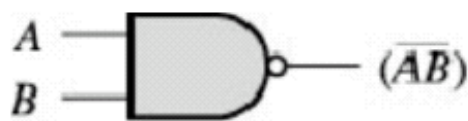
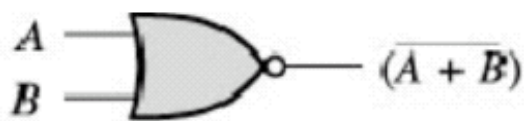
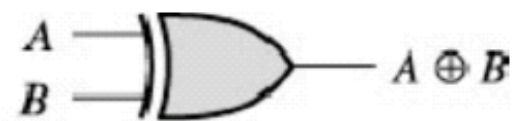
- **NAND** è una funzione logica a 2 o più ingressi, che restituisce in uscita la funzione **AND negata** degli ingressi, quindi è ‘0’ quando tutti gli ingressi sono ‘1’ (il cerchietto all’uscita dell’AND indica che l’uscita va invertita)
- **NOR** è una funzione logica a 2 o più ingressi, che restituisce in uscita la funzione **OR negato** degli ingressi, quindi è ‘0’ quando almeno uno degli ingressi è ‘1’
- **XOR** è una funzione logica a 2 ingressi (detta **OR-esclusivo** e/o **somma modulo-due**), che restituisce ‘1’ quando gli ingressi sono diversi, ‘0’ quando sono uguali

A, B → $(\overline{AB})$

(a) NAND gate

A, B → $(\overline{A + B})$

(b) NOR gate

A, B → $A \oplus B$

(c) XOR gate

Principali funzioni logiche

Espressione algebrica

Simbolo grafico

Tabella di verità

NOT

$Z=X'$

X	Z
0	1
1	0

OR

$Z=X+Y$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

AND

$Z=X \cdot Y$

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Le operazioni di somma e prodotto logico possono essere estese a più di 2 variabili, in generale per n variabili di ingresso si ha:

$Z = A + B + C + \ldots$ $\quad\quad$ $Z = A \bullet B \bullet C \bullet \ldots$

NOTA : fare esercizi 3.1, 3.3, 3.5 del libro pag. 60 e seg.

Principali funzioni logiche

NOR — $Z=(X+Y)'$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

NAND — $Z=(X \cdot Y)'$

X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

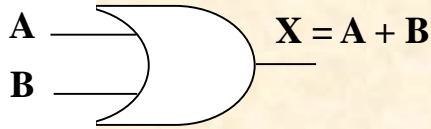
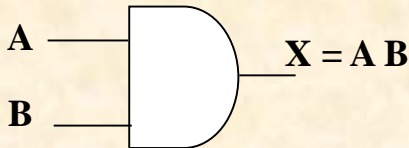
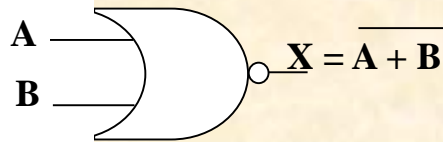
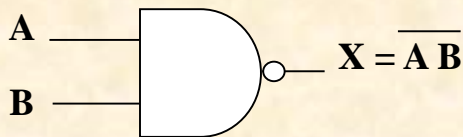
XOR — $Z= X \cdot Y' + X' \cdot Y$

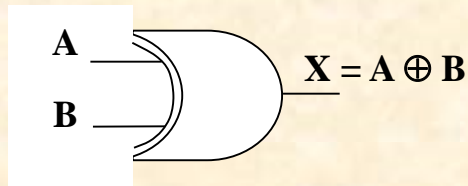
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

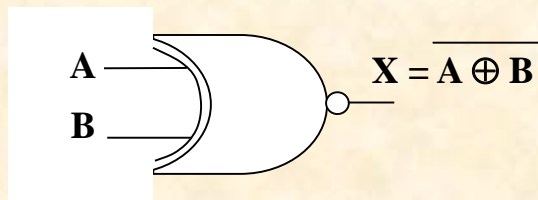
XNOR — $Z=X' \cdot Y'+X \cdot Y$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

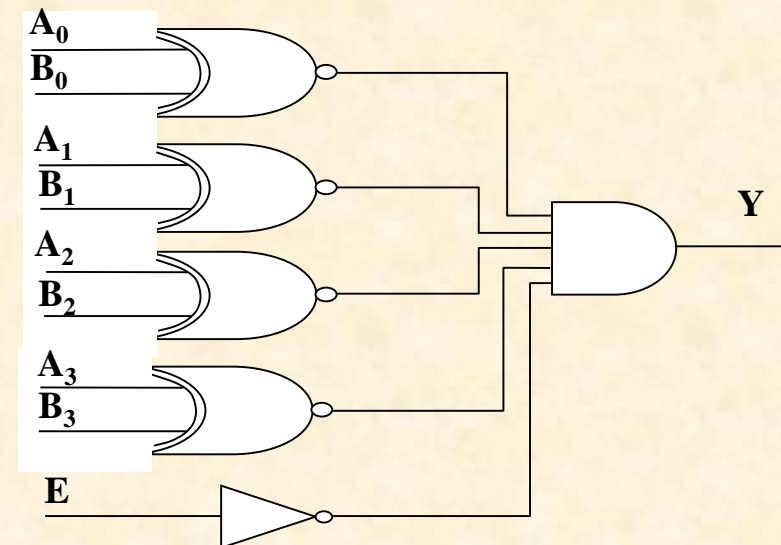
Circuiti di OR ESCLUSIVO e di NOR ESCLUSIVO – VEDI LIBRO PAG. 151- 157.

NOTA: fare esercizi figura 3.15 e 3.8 (pag 75) 3.10 e 3.11 del libro.

È sufficiente che un ingresso sia 1 per avere X = 1.
Viceversa X = 0 solo se tutti gli ingressi sono 0.

È sufficiente che un ingresso sia 0 per avere X = 0.
Viceversa X = 1 solo se tutti gli ingressi sono 1.

L'uscita X è 1 solo se tutti gli ingressi sono 0. E' sufficiente che un solo ingresso sia 1 per avere X = 0. Equivale ad eseguire l'OR tra A e B ed invertire il risultato.

L'uscita è 0 solo se tutti gli ingressi sono 1. E' sufficiente che almeno un ingresso sia 0 per avere X=1. Equivale ad eseguire l'AND tra A e B ed invertire il risultato.

EXOR

L'uscita è 1 solo se gli ingressi sono diversi fra loro (or esclusivo). Viceversa l'uscita è 0 se gli ingressi sono uguali fra loro (comparatore di uguaglianza con uscita attiva bassa).

EX-NOR

L'uscita è 1 solo se gli ingressi sono uguali fra loro (comparatore di uguaglianza con uscita attiva alta). Viceversa l'uscita è 0 se gli ingressi sono diversi fra loro.

Comparatore di uguaglianza di 2 parole digitali A e B di n bit, con bit E di disabilitazione.

Abilit. E	Ingressi A_n	B_n	Uscita Y
1	x	x	0
0	A = B		1
0	A ≠ B		0

TEOREMA DI DE MORGAN

- Data la funzione binaria F di più variabili booleane (A, B, C, ..., N) vale la seguente identità:

$$\overline{F(A, B, C, \ldots, N, +, *)} = F(\overline{A}, \overline{B}, \overline{C}, \ldots, \overline{N}, *, +)$$

dove nella funzione al secondo membro si è sistematicamente sostituita ogni variabile con il suo complemento e si sono scambiati fra loro i simboli delle operazioni di somma e di prodotto.

- Ad esempio:

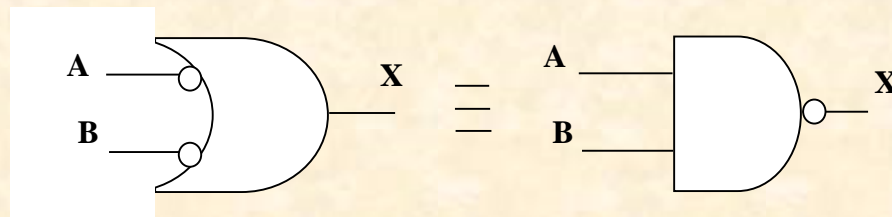
$$\overline{A+B} = \overline{A} * \overline{B}$$

$$\overline{A*B} = \overline{A} + \overline{B}$$

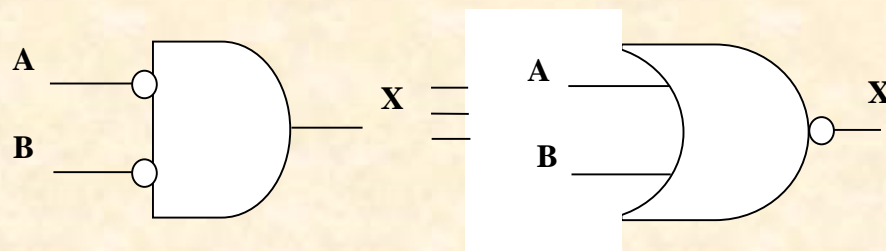
Fare considerazione su circuiti equivalenti secondo il teorema di DE MORGAN vedi libro figure 3.26 e 3.27 pag. 87 ed esercizio 3.17.

Circuiti equivalenti secondo il teorema di De Morgan

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

A
B
X
≡
A
B
X

Simbolo alternativo per la porta NAND: porta OR con ingressi attivi bassi ed uscita attiva alta.

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

A
B
X
≡
A
B
X

Simbolo alternativo per la porta NOR: porta AND con ingressi attivi bassi ed uscita attiva alta.

Rappresentazione di funzioni logiche

- Una qualsiasi funzione logica può essere rappresentata in svariati modi.
  - **Tabella di verità**: la tabella di verità ha tante righe quante sono le possibili combinazioni degli ingressi e per ogni riga viene indicato il valore della funzione
  - **Espressione logica**: la funzione è rappresentata per mezzo di un'espressione algebrica contenente le variabili di ingresso e gli operatori logici di base
  - **Mappe di Karnaugh**: rappresentazione grafica basata sulla visualizzazione delle combinazioni di ingressi per cui la funzione vale 1 (o 0), utilizzata per la minimizzazione della funzione stessa
  - **Schematico**: rappresentazione grafica per mezzo di simboli

- Prima forma canonica: somma di prodotti, ovvero somma di mintermini: $f(x_1,\ldots,x_n)=m_1+m_2+\ldots+m_k$
- Seconda forma canonica: prodotto di somme, ovvero prodotto di maxtermini: $f(x_1,\ldots,x_n)=M_1M_2\ldots M_k$

Somma canonica

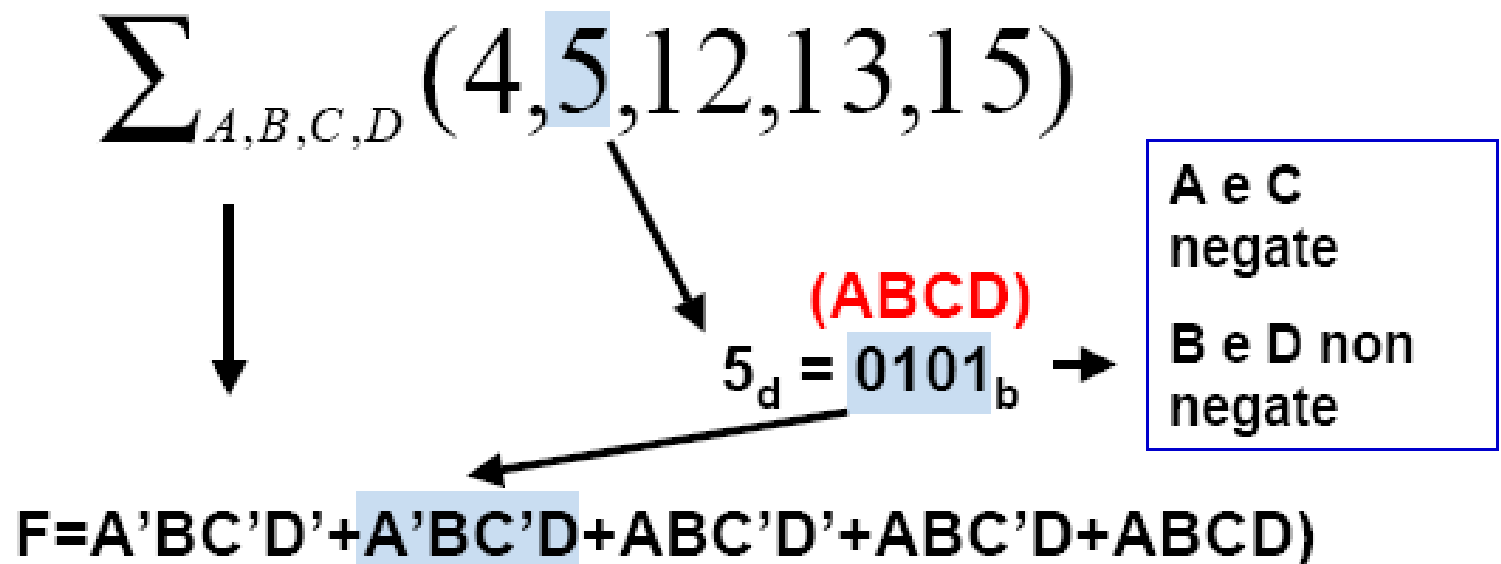
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

- Un altro modo, molto compatto, per rappresentare una funzione logica è quello della sua **forma canonica** (somma o prodotto)
- Nel caso della somma si selezionano solo le righe che contengono 1 e si rappresentano con un numero che corrisponde al numero di riga

$$\sum_{A,B,C,D}(4,5,12,13,15)$$

Le forme canoniche descrivono circuiti a due *livelli* (senza contare gli eventuali NOT necessari per negare le variabili di ingresso). **Livello:** numero di livelli di porte logiche che i segnali devono attraversare per propagarsi dagli ingressi all'uscita.

Somma canonica

- Avendo la somma canonica è possibile scrivere facilmente l'espressione algebrica: si fa una somma di termini ciascuno dei quali è il prodotto delle variabili stesse prese però negate se il bit corrispondente nel codice binario della riga è 0

Funzione logica espressa come somma di mintermini

Prodotto canonico

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

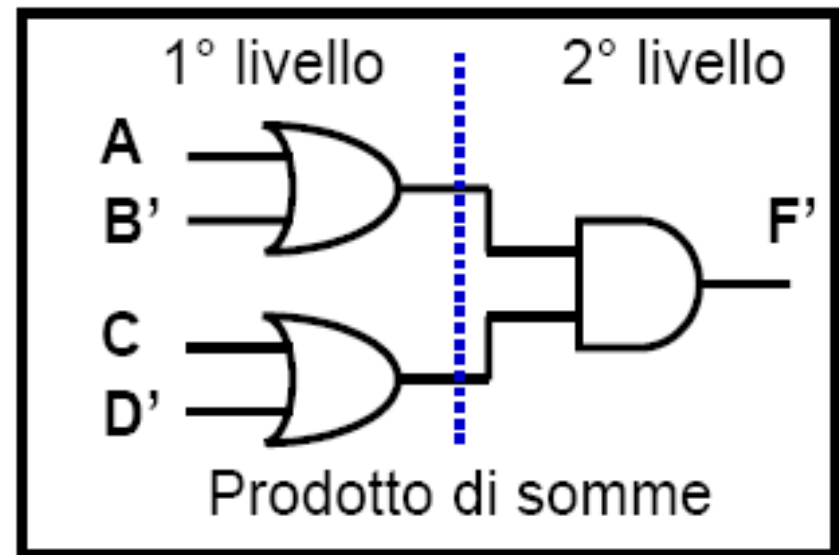
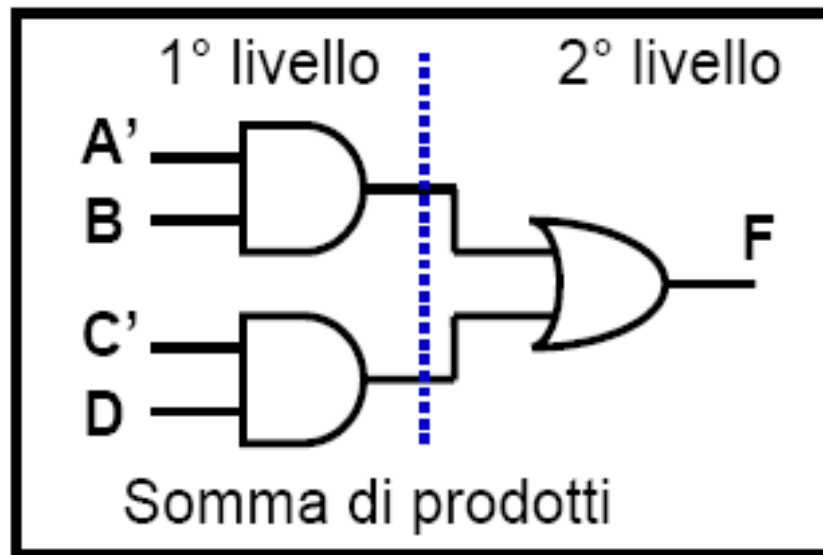
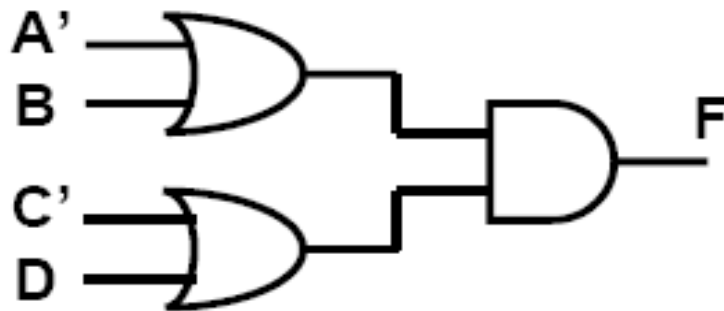
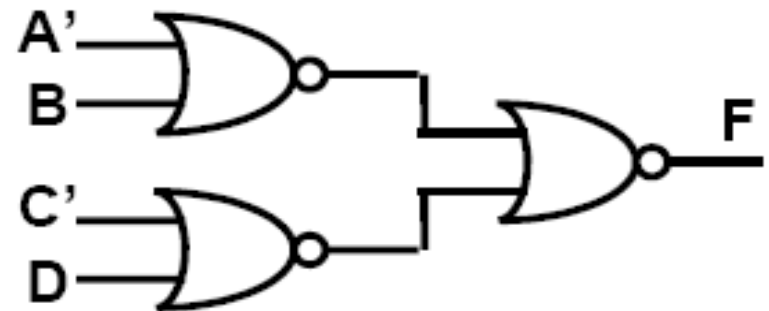
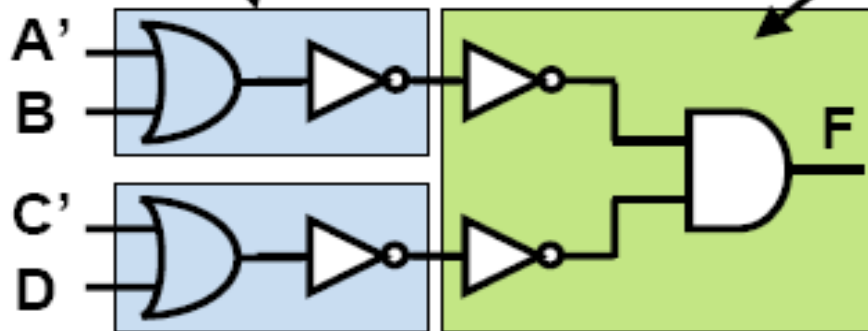
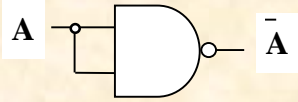
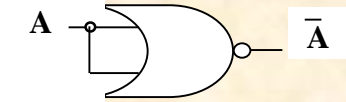
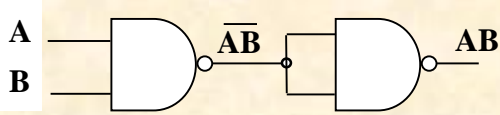
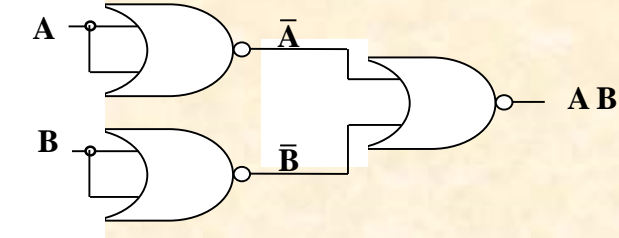
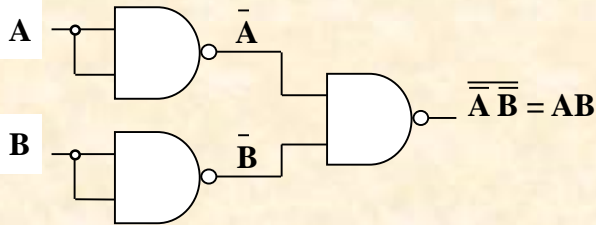
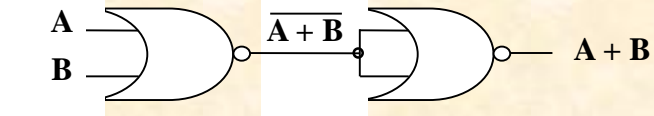
- Si selezionano le righe contenenti 0
- Nel passare alla espressione algebrica si usa la convenzione opposta (la variabile è negata se c'è un 1 nel codice della riga)

$$\prod_{A,B,C,D}(0,1,2,3,6,7,8,9,10,11,14)$$

F=(A+B+C+D) (A+B+C+D') (A+B+C'+D) (A+B+C'+D') (A+B'+C'+D) (A+B'+C'+D') (A'+B+C+D) (A'+B+C+D') (A'+B+C'+D) (A'+B+C'+D') (A'+B'+C'+D)

Funzione logica espressa come prodotto di maxtermini

Implementazione di funzioni logiche

- E' dimostrabile che qualsiasi funzione logica può essere implementata con i soli operatori di somma, prodotto e negazione e con solo 2 livelli di logica. Ossia con somme di prodotti o prodotti di somme.

Implementazione con operatori NAND

Per il teorema di DeMorgan è possibile trasformare la somma di prodotti in modo da avere solo operatori NAND

Vedi libro, paragrafo 3.12 – UNIVERSALITA' DI PORTE NAND e NOR pag. 88 – 92 con esercizio 3.18; e paragrafo 3.13 RAPPRESENTAZIONE ALTERNATIVE DELLE PORTE LOGICHE PAG. 93-97.

Universalità delle porte NAND e NOR

Qualsiasi espressione logica può essere implementata con l'utilizzo di sole porte NAND o NOR; ciò è dovuto al fatto che le funzioni logiche AND, OR e NOT si possono ottenere con combinazioni di sole porte NAND e NOR.

	NAND	NOR
NOT	A → $\overline{A}$	A → $\overline{A}$
AND	A, B → $\overline{AB}$ → AB	A → $\overline{A}$; B → $\overline{B}$ → A B
OR	A → $\overline{A}$; B → $\overline{B}$ → $\overline{\overline{A}\,\overline{B}}$ = AB	A, B → $\overline{A + B}$ → A + B

RAPPRESENTAZIONE ALTERNATIVE DELLE PORTE LOGICHE

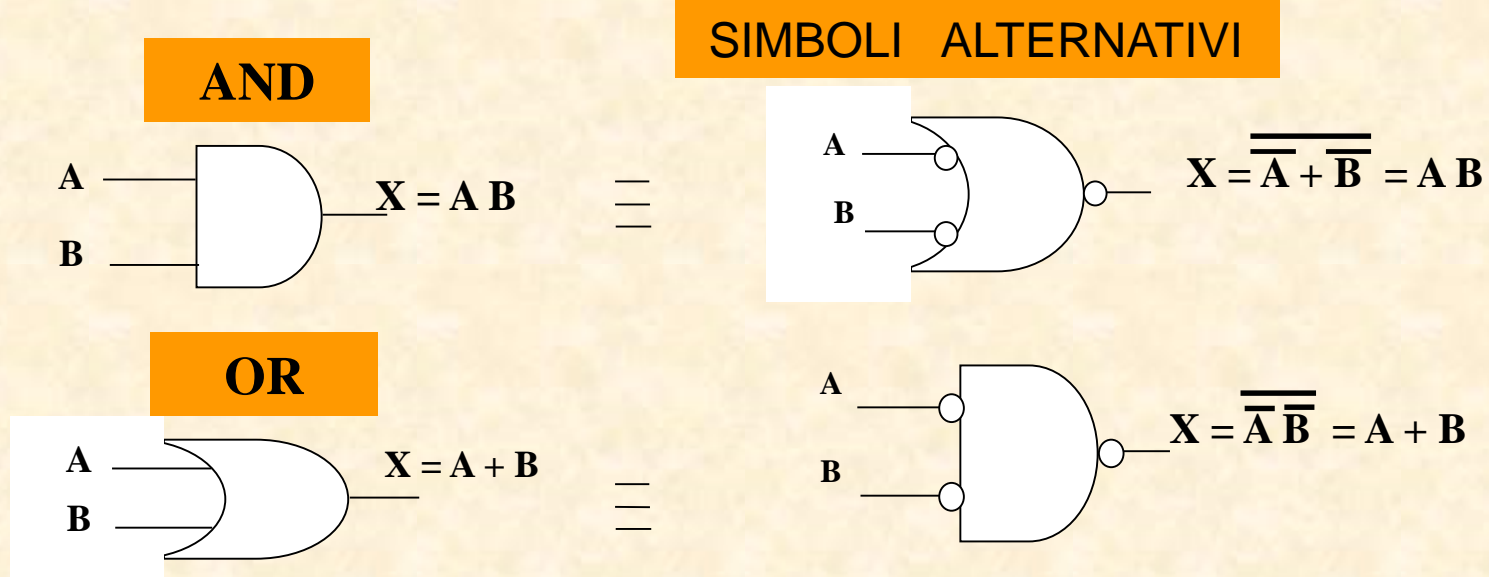
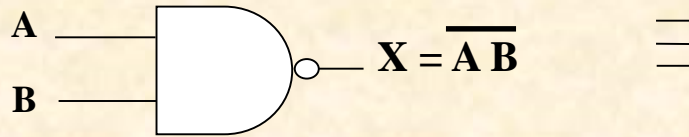
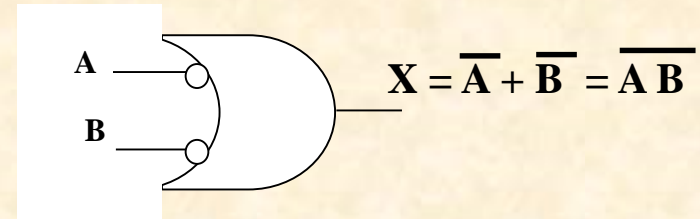
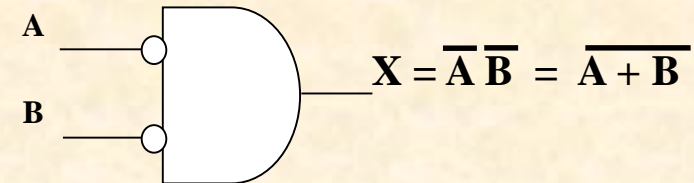
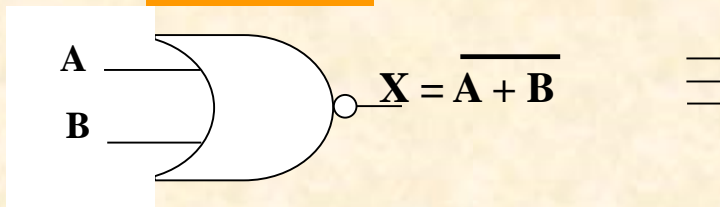
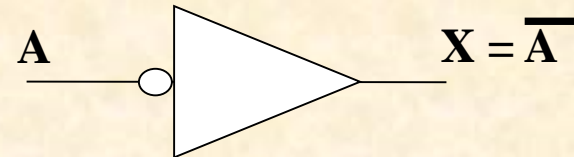
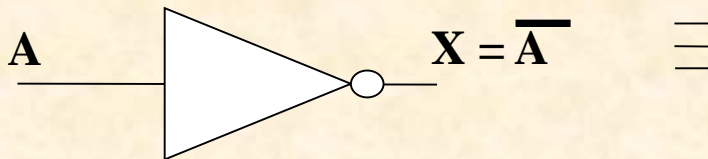
Per ciascuna porta logica il simbolo alternativo è ottenuto dal simbolo standard tramite le seguente due azioni:

- Inversione di ogni ingresso e di ogni uscita del simbolo standard. Ciò si ottiene ponendo un cerchiolino sulle linee di ingresso e di uscita che ne sono sprovviste e rimuovendo i cerchiolini dalle linee che già lo hanno.

- Sostituzione della porta AND con OR e della porta OR con AND.

≡

≡

≡

- Le equivalenze sono valide per porte con un numero qualsiasi di ingressi.

- Nessuno dei simboli standard ha dei cerchiolini sui propri ingressi, al contrario di tutti i simboli alternativi.

- I simboli standard ed alternativi per ciascuna porta rappresentano il medesimo circuito elettronico.

MAPPE DI KARNAUGH

- Le mappe di Karnaugh sono strumenti di rappresentazione e minimizzazione delle somme di prodotti con un numero di termini minore o uguale a quattro. Ciascuna casella della mappa di Karnaugh corrisponde ad un mintermine della funzione; sui bordi della mappa sono infatti rappresentate le variabili che hanno la funzione di identificare il mintermine contenuto nella casella. Nella disposizione di due o più variabili nella stessa riga o colonna è necessario che tra una riga (o colonna) e la successiva si modifichi il valore di una variabile alla volta.
- Osservando la disposizione e l'adiacenza dei valori '1' contenuti nelle caselle della mappa è possibile costruire sottoinsiemi (2*1), (2*2), (4*1) ecc. nei quali la stessa variabile è negata e non negata. Questo significa che il valore della suddetta variabile, in corrispondenza di determinati valori delle altre variabili, è ininfluente ai fini del calcolo del valore della funzione.
- E' quindi possibile semplificare la funzione.

Mappe di Karnaugh

Il metodo delle mappe di Karnaugh è una tecnica grafica di minimizzazione; si devono disegnare mappe con N caselle per ogni combinazione delle n variabili di ingresso, quindi risulta **$N = 2^n$** .

Funzione ***f*** a 2 variabili di ingresso A e B, mappa con 4 caselle.

Funzione ***f*** a 3 variabili di ingresso A B e C, mappa con 2^3 = 8 caselle.

A \ BC	00	01	11	10
0				
1				

Funzione ***f*** a 4 variabili di ingresso A B C D, mappa con 2^4=16 caselle pari a tutte le possibili combinazioni delle variabili di ingresso.

AB \ CD	00	01	11	10
00				
01				
11				
10				

Le mappe di Karnaugh con 5 variabili sono ottenute dall'unione di 2 mappe a 4 variabili con E=0 ed E=1.

Mappe di Karnaugh

- La rappresentazione per mezzo di mappe di Karnaugh permette di trovare l'implementazione in ***forma minima***.

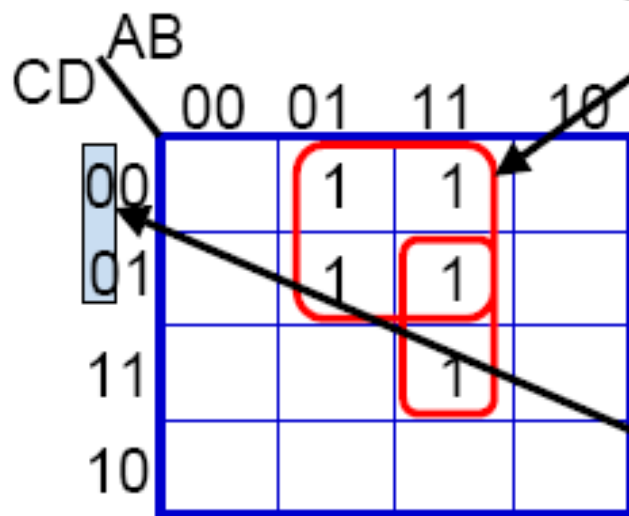
Mappe di Karnaugh

- Per minimizzare la funzione si *coprono* tutte le caselle contenenti gli 1 con gli implicanti più grandi possibili (ossia con riquadri contenenti un numero di caselle che sia una potenza di 2 e che contengono solo 1).
- Dopodiché la forma su due livelli di logica della funzione è ottenuta come somma dei prodotti rappresentati dagli implicanti

Implicante di una funzione data è il prodotto di letterali.

Mappe di Karnaugh

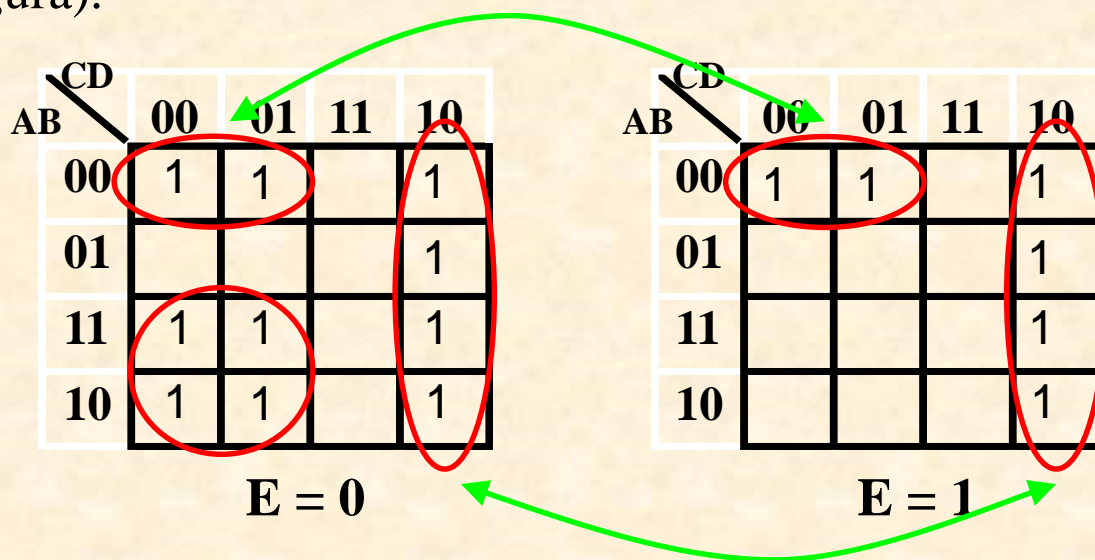
Nel termine prodotto non deve comparire la variabile che cambia di valore nel passaggio da una cella a quella adiacente.

Per ogni implicante si mettono solo le variabili che non cambiano all'interno dell'implicante stesso

Un implicante con 2^k caselle ha N-k variabili

Se dentro l'implicante la variabile è pari a 1 si mette la variabile stessa, altrimenti si mette la variabile negata

Mappe di Karnaugh

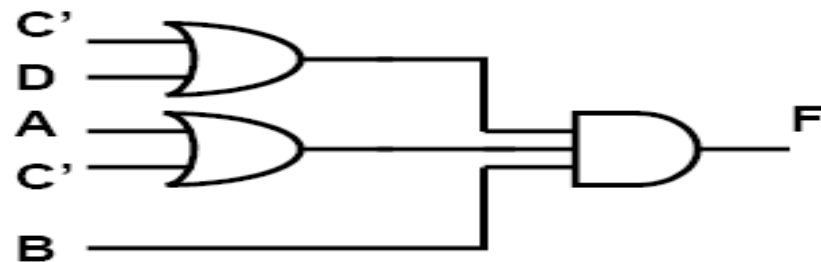
Le mappe di Karnaugh con 5 variabili sono ottenute dall'unione di 2 mappe a 4 variabili con E = 0 ed E = 1.

Attenzione: risultano adiacenti anche le celle nella stessa posizione (stessa configurazione per gli ingressi ABCD) appartenenti alle 2 mappe (con E=0 e E=1) (vedi esempio in figura).

Per determinare la forma minima della funzione, essendo il costo del circuito dipendente dal numero di porte logiche, si devono coprire gli 1 della mappa col ***minor numero di sottocubi più grandi possibile***.

$$Y = \overline{A}\,\overline{B}\,\overline{C} + C\,\overline{D} + A\,\overline{C}\,\overline{E}$$

Mappe di Karnaugh

- Per ottenere la funzione in termini di prodotto di somme bisogna modificare la rappresentazione:
  - Si mettono zeri nelle caselle corrispondenti a combinazioni d'ingresso per cui la funzione è 0
  - Si coprono con implicanti contenti solo zeri
  - Nell'espressione algebrica dell'implicato si mette la variabile stessa se essa compare col valore 0 altrimenti si mette la variabile negata se compare col valore 1

- Nel caso della funzione precedente

Vedi libro, paragrafo *Mappe non completamente specificate* pag. 97-100 ed *Alee statiche* paragrafo 8.5 pag. 100-103.

Mappe non completamente specificate

In alcune reti combinatorie non è necessario definire la funzione booleana per tutte le possibili configurazioni delle variabili di ingresso in quanto alcune di esse non si presenteranno mai in ingresso.

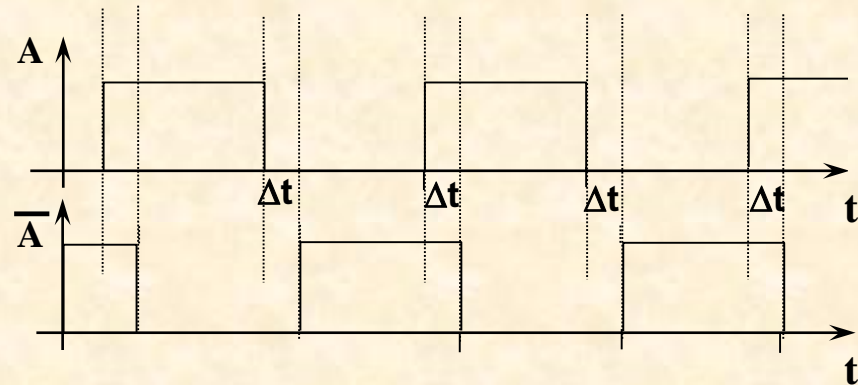
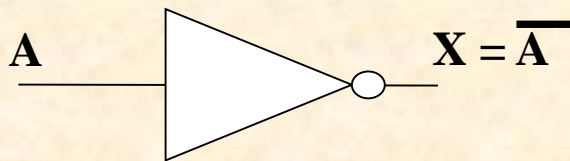
Nella mappa, in corrispondenza alle configurazioni non utilizzate, possiamo segnare per *f* una X il che significa che attribuiamo alla funzione *f* un valore logico 0 oppure 1 indifferentemente secondo le esigenze di ricopertura della mappa.

Questo è possibile perché in ingresso non si presenteranno queste configurazioni in ingresso e quindi non è necessario prevedere alcun valore logico da assegnare ad *f* .

Le condizioni di indifferenza X sono utili in quanto possono rendere i sottocubi più grandi possibile e quindi minimizzare ulteriormente la funzione logica.

$$F = \overline{B}\,\overline{C}\,\overline{D} + A\,D$$

Alee statiche in reti combinatorie

In realtà alle porte logiche ed ai dispositivi elettronici sono associati dei ritardi di propagazione che possono provocare dei malfunzionamenti dell'intera rete combinatoria soprattutto all'aumentare della frequenza di lavoro.

Ad esempio per una porta NOT la risposta dell'uscita avviene in ritardo di Δt rispetto al segnale di ingresso.

Quindi il problema delle alee, che non esiste nella teoria considerando ideali i dispositivi e le porte logiche, deve essere affrontato e risolto per poter passare alla realizzazione circuitale facendo uso di dispositivi e porte logiche reali.

Alee statiche in reti combinatorie

Data la mappa di Karnaugh in figura, la forma minima risulta:

$$F = \overline{A}B + AC$$

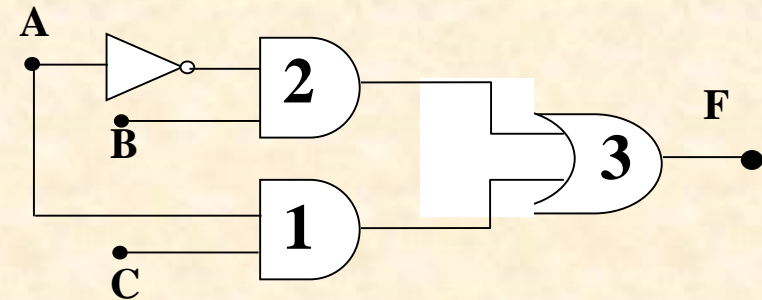
Un'alea statica si verifica nella commutazione di un ingresso qualora l'uscita dipenda direttamente da esso e dal suo valore negato. Si può avere un'alea statica di **tipo 0** (zero spurio o indesiderato dell'uscita della rete combinatoria) o di **tipo 1** (uno spurio o indesiderato).

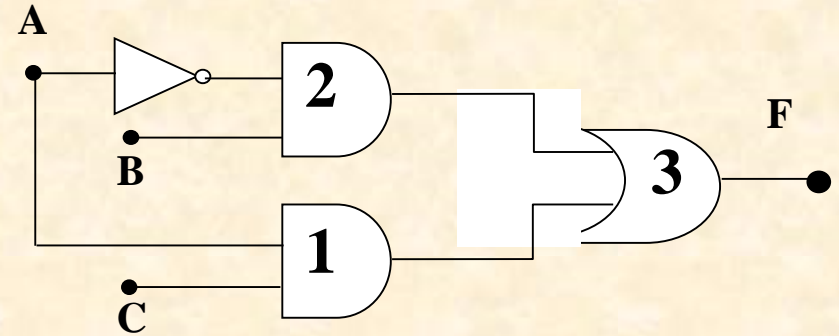
Le alee statiche possono essere rilevate dalle mappe di Karnaugh quando si presentano due 0 (o due 1) adiacenti non raggruppati. In altre parole si hanno alee statiche di tipo 0 (1) tutte le volte che sulla mappa si passa da un 1 (0) di una cella a quello di un'altra cella adiacente cambiando la ricopertura; infatti in questo caso l'1 (0) deve essere fornito da porte diverse che possono non commutare contemporaneamente.

A \ BC	00	01	11	10
0			1	1
1		1	1	

Vi sono due 1 adiacenti non raccolti dallo stesso sottocubo. L’alea statica di tipo 0 si presenterà dunque nel passaggio dalla combinazione degli ingressi 011 (per la quale F=1) a quella 111 (per la quale F=1) oppure nel passaggio inverso.

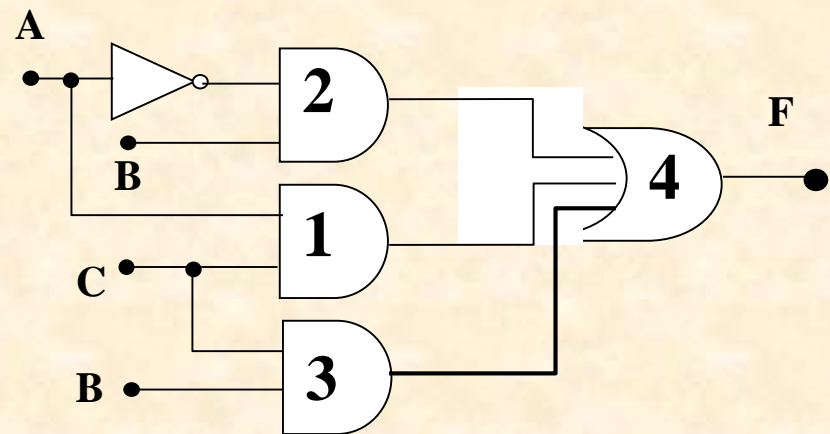
$F = \overline{A}B + AC$

Per stabilire ciò occorre ragionare sul circuito combinatorio relativo alla funzione logica F.

Consideriamo la situazione A=B=C=1, $\overline{A}$=0; all’uscita della porta AND-1 ci sarà un 1 e quindi F=1. All’istante t_o si verifica la transizione di A da 1 a 0 (combinazione degli ingressi da 111 a 011); la porta NOT presenta un ritardo pari a Δt (alcuni nanosecondi) per cui la sua uscita $\overline{A}$ commuta da 0 a 1 con ritardo Δt. L’uscita della porta AND-1 commuta da 1 a 0 senza ritardo mentre l’uscita della porta AND-2 da 0 a 1 con ritardo Δt. Per un breve intervallo di tempo entrambe le uscite delle porte AND sono = 0 e quindi F = 0 (alea statica).

Consideriamo la situazione A=0, B=C=1, $\overline{A}$=1; all'uscita della porta AND-2 ci sarà un 1 e quindi F=1.

All'istante t_o si verifica la transizione di A da 0 a 1 (combinazione degli ingressi da 011 a 111); la porta NOT presenta un ritardo pari a Δt (alcuni nanosecondi) per cui la sua uscita $\overline{A}$ commuta da 1 a 0 con ritardo Δt. L'uscita della porta AND-1 commuta da 0 a 1 senza ritardo mentre l'uscita della porta AND-2 da 1 a 0 con ritardo Δt. Per un breve intervallo di tempo entrambe le uscite delle porte AND sono = 1 e comunque la porta AND-1 (la cui uscita è 1 senza ritardo) garantisce F = 1 (non si ha alea statica).

Per eliminare l'alea statica, bisogna raccogliere i due 1 adiacenti rendendo la funzione ridondante.

$$F = A\overline{B} + AC + BC$$

Rete combinatoria NON minima senza il problema dell'alea statica di tipo 0

Se la combinazione degli ingressi cambia da 111 a 011 (transizione che produceva l'alea statica di tipo 0), l'uscita della porta AND 3 (termine prodotto BC) è sempre pari ad 1 indipendentemente dall'ingresso A che cambia garantendo un 1 costante in ingresso alla porta OR e dunque F = 1 (il problema dell'alea di tipo 0 è stato risolto con l'aggiunta della porta AND 3).

Possiamo concludere che una rete combinatoria non presenta alee statiche se ogni coppia di 1 (o di 0) adiacenti è racchiusa da almeno un sottocubo comune.

La necessità di eliminare le alee (malfunzionamento del circuito logico) ed aumentare l'affidabilità del circuito aggiunge porte logiche ridondanti (rete non minima) ed aumenta quindi la complessità ed il costo del circuito.